Assessing the Efficacy of Integrating Computer Science, Math, and Science in a Middle School Sphero Robotics Summer Program

Talese Simley
Psychology
Spelman College
Atlanta, GA 30314, USA
tsimley@scmail.spelman.edu

Naja A. Mack
Computer Science
Morehouse College
Atlanta, GA 30314, USA
naja.mack@morehiouse.edu

Tristian Pittman
Computer Science
Morehouse College
Atlanta, GA 30314, USA
tristian.pittman@morehouse.edu

Chassidy Cook
Computer Science
Morehouse College
Atlanta, GA 30314, USA
chassidy.cook@morehouse.edu

Robert Cummings
Computer Science
Morehouse College
Atlanta, GA 30314, USA
robert.cummings@morehouse.edu

Dekita Moon
Computer & Information Science & Engineering
University of Florida
Gainesville, FL 32611, USA
dekita@ufl.edu

Kinnis Gosha
Computer Science
Morehouse College
Atlanta, GA 30314, USA
kinnis.gosha@morehouse.edu

***Abstract*—There have been many discussions on the significant lack of representation of minorities in computing. Research suggests that an attributing factor to this deficit is a lack of exposure to computing, computing careers, and underrepresented role models. Introducing underrepresented minority students to hands-on computer science activities at an early age is believed to improve students' attitudes in computing. Increasing students' computing attitudes may help encourage participation and minimize deficiencies in retaining underrepresented minority computing students. An interactive summer program was implemented to teach computing to middle-school-aged underrepresented minority students through a unique curriculum using Sphero robotic balls and math and science concepts. Students were assessed to determine the programs' effectiveness in increasing computing attitudes and academic performance of math and science. Results showed that the intervention significantly improved students' academic performances and their confidence in deciding a future career path. Further evaluation is needed to explore longitudinal effects of the varying levels of math, science, and computing rigor used within this intervention. Findings from this study provide recommendations for how to implement computing practices within the existing math and science curriculum in prospective initiatives.**

***Keywords*—computer science, outreach, math, science, robotics, middle school**

I. Introduction

Introducing computer programming to students at an early age aids in the development of transferable skills that will bridge the racial and educational divide [1]. In 2017, only 8% of African Americans, who were enrolled at a 4-year institution, graduated with a degree in computer science. The numbers only get smaller as these students matriculate through graduate school and/or pursue a doctorate [2]. With the growing pervasiveness of technology in our daily lives, coding is becoming a fundamental skill for learning; comparable to skills such as reading, writing, and arithmetic. Over the last several years, 58% of all new jobs in STEM are in computing, but only 10% of students that majored in STEM fields received a degree in computer science [3]. The development and retention of programming skills begin with teaching and providing tasks that encourage more complex diagnostic thinking.

Coding has been proven to develop skills in math, reading, logic, computational-thinking, and applied sciences [4], [5]. These authors believe there is a way to help students understand the underlying computing skills in conjunction with how they process the information on their own. Exposing the students to rigorous and relevant adaptive learning may lead to higher retention of student engagement/involvement in computing. Literature suggests that robotics is becoming a more effective way to engage students with coding and programming. Additionally, robotics helps reinforce soft skills such as teamwork, critical and creative thinking, problem-solving, and algorithmic patterned thinking [6]. Robotic-based programs are often effective at retaining a students' computing knowledge for those who have never programmed before.

The participants in this study are unique because underserved minority middle schoolers are often overlooked and underrepresented when it comes to computing research. Student engagement in computer programming manifests in behaviors that are observable and recurring in middle school students [7], [8]. These behaviors result in:

- Engaging in often short-lived, intense interests
- Preferring interactions with their peers
- Preferring active to passive learning

Previous studies have shown that adolescent students retain

978-1-7281-7172-2/20$31.00 ©2020 IEEE

computing knowledge more effectively when computing is taught through a visceral, “hands-on learning experience” [9]. Implementing teaching strategies, such as problem-based learning within a STEM curriculum, may enhance students’ desire to understand the world around them and engage them in classroom instruction [10]. Skills like problem-solving literacy, creativity, and motivation are positively influenced when children access technology in their learning environments. Also, using technology as an instructional tool enhances children’s learning and educational outcomes [11]. More specifically, introducing computer science to underrepresented minorities at an early age can influence more positive attitudes towards computing; consequently, attracting more diverse talent to the field of technology and computing [12].

II. BACKGROUND/RELATED WORK

While some minority groups are well represented in technology, others are almost nonexistent, such as African Americans [1], [3], [5], [13]. Connecting with the next generation of minority talent means early introduction, easily accessible resources, and providing a hands-on learning experience. An early introduction to computing may increase the participation of underrepresented groups in the tech field because experiences and exposure in childhood affect desired careers in adulthood [13], [14]. To bridge the gap, which may result in upward mobility, it is necessary to effectively engage racial and ethnic minority groups and other vulnerable and underserved populations, with the help of outreach/bridge programs.

Many programs, similar to ours, cater to fostering awareness and passion for computer science amongst adolescents and/or minorities. An example study was conducted at the Department of Computer and Cyber Sciences at the United States Air Force Academy. They examined how accessibility, early introduction, and hands-on experiences increased students’ computing attitudes. They tested a proposed introductory programming curriculum that aligned with the College Board’s Advanced Placement Computer Science Principles (AP CSP) course for the Sphero SPRK+. The SPRK+ is the predecessor to the newer Sphero Bolt model used in this study. The purpose of the study at the Academy was to increase student engagement and to “motivate and facilitate the effective learning of introductory programming and problem-solving skills” by using robotics [15].They tailored each lesson plan utilizing a problem-solving methodology called UDIT (Understand – Design – Implement – Test). The UDIT methodology provides specific tasks, goals, and techniques for each of the phases. In addition to similar pedagogies, the curriculum at the academy covered topics such as, but not limited to, the following:

- **Introduction to Programming the SPRK+ using Blocks:** To introduce the students to creating, editing, and running programs in the Blocks canvas, this activity used the Sphero Block activities to teach the Roll, Stop, Delay, Spin, Main LED, Speak, Fade, and Strobe.
- **Selection, Loop Forever, Sensor Data, and Comparators:** To introduce selection and iteration control logic as well as yaw, roll, and pitch sensor readings from the SPRK+ gyroscope.
- **Gyroscopes, Normalization, Lights, and Math Functions:** To develop an understanding of what a gyroscope does and how to normalize data from one range to another.
- **Variables, Operators, Loop Until, Randomization, and Haptic Feedback:** To extend the students’ understanding of key programming concepts and techniques, this lesson used the Sphero Blocks 4 activity, from the Sphero website, to teach the concept and use of variables, math operators, loop until construct, and random numbers generation. The Blocks 4 activity had students develop a Hot Potato game which introduced them to haptic feedback by using raw motor settings to vibrate the SPRK+.

The objective of this effort was targeted toward the development of a theme-based curriculum using the Sphero SPRK+ that would effectively achieve the required learning while increasing student interest and engagement, similar to our own course’s goal. 63.2% of their students agreed that the activities were interesting and engaging and are helpful for learning how to program and 26.3% of their students strongly agreed that the activities were interesting, engaging and helpful for learning how to program [15].

These results support the idea that younger students are prone to lack the patience and abstract thought necessary to complete activities such as programming [4]. On the other hand, waiting until college to address the issue of minorities in computer science is non-beneficial. Therefore, using robotics early on can provide experience to improve computer science grades in the future [1]. The use of robotics can provide a visceral, “hands-on learning experience” for students who have never programmed before; this is essential in the retention of the students [16]. This is why outreach programs introduce minorities to computing and the tech industry. Outreach programs create an environment where students can learn and retain. Implementing teaching strategies, like problem-based learning within a STEM curriculum, may enhance students’ desire to understand the world around them and engage them in classroom instruction. This mitigates the susceptibility of students becoming impatient and creates an environment that supports abstract thinking [6]. Skills like problem-solving literacy, creativity, and motivation are positively influenced when children access technology in their learning environments. Also, using technology as an instructional tool enhances children’s learning and educational outcomes [7]. More specifically, introducing computer science to minorities at an early age will not only attract more diverse talent to the field of technology and computing, but will also aid with the retention of said children [17].

III. PROGRAM OVERVIEW

The research team was given an opportunity to participate in a summer program dedicated to the enrichment of gifted and high-achieving students within a local public school district. The goal of this program was to expose the students to the world of coding while integrating relevant science and math concepts. The program provided various courses that were geared towards STEM and fine arts. In this course, rising 6th - 8th graders were introduced to the world of coding, via the Sphero Bolt which is an app-enabled robot. The Sphero Bolt application allows users to code in three separate ways: generate drawings that the robot executes, drag and drop blocks of code to create programs, or write JavaScript programs for execution. For this course, the students began creating programs with the drawing function so they could learn how to manipulate the robot. After becoming familiar with Sphero's functionalities, students were restricted to only using the "drag and drop" method with blocks of code to complete activities for the remainder of the summer program.

The Sphero team (the instructors and observers) began prepping for this course by developing and practicing lesson plans three weeks before the summer program began. In the preparation phase, the Sphero team created PowerPoint slides, worksheets, quizzes on Kahoot! (a game-based learning platform), and Sphero Bolt activities for each lesson. These instructional tools were a combination of projects found on the Sphero website, the Atlanta Public Schools curriculum, and original ideas created by the research team. The content for each project on the Sphero website included instructional materials that were customized to lessons along with the required materials to complete the activity. Materials used during the program included tape, protractors, ramps, string, wooden sticks, and a pool of water.

The instructors practiced and reviewed the lessons and activities multiple times with other members of the lab. During these practice sessions, the instructors presented their instructional materials and received feedback on the content, rigor, appropriateness, and flow of the presentation. Each presentation was followed by a brief review period; identical to the review period students participating in the workshop would experience. The review sessions included worksheets or "Kahoot!" activities to recap the daily lesson. These instructional rehearsals concluded with the team reviewing the Sphero activity for the day in which feedback was obtained and unforeseen challenges were discussed.

A. Structure

The course lasted for four weeks starting on Monday and ending Friday, from 11:30 a.m. to 3:30 p.m. As stated previously, the goal of this program was to expose the students to the world of coding while integrating relevant science and math concepts. The first week of the program focused on Sphero basics, the second focused on math concepts, and the third week focused on science. The students were assigned to teams and each team was assigned an observer to help them during the classes. Each day, the students were introduced to a new topic with a corresponding activity using the Sphero robots. At the beginning of each week, students were given an assessment with content related to the lesson being taught that week. The same assessment was administered at the end of the week. Students completed all the assessments before going to lunch to maintain their interest and limit interruptions and student distractions. Conducting all assessments before lunch also gave the instructors time to briefly look over the results and alter the lesson plan as needed to ensure that the students were gaining an understanding of the lesson content.

Lessons began immediately after their lunch. Each lesson included a PowerPoint and a brief review session, which involved completing a worksheet or a Kahoot! quiz. Overall, the lesson and review period lasted between 45-60 minutes. The lesson and review period was followed by an activity. Each Sphero activity maintained relevance to reinforce and build upon the topic of that day. For example, the review session for the angles math lesson consisted of students creating and identifying angles and shapes on a worksheet. The activity was then followed by students recreating the angles using the Sphero balls. At the end of each class, notes and comments were collected from students with a brief reflection period. This gave students a chance to voice their thoughts about the lesson and activity for that day.

B. Sphero Basics

The first week consisted of ice breaker challenges, grouping the students, and teaching the young coders how to utilize and control the Sphero balls. For the control activities, the students learned how to aim and drive the Sphero Bolt robot. Then, they moved into block coding. Activities, such as bowling and soccer were integrated with each lesson to ensure that everyone had grasped the concept. Since time was limited, students were taught the foundational necessities, including but not limited to, changing the light, adjusting the speed, and aiming the ball. Students also participated in a reflection period at the end of each day, where the most insight was gained on their comfortability with the functions of the robot. An analysis of the reflection responses showed that most of the students felt comfortable controlling the ball within the first day and a half, with only one student having prior experience with Sphero. After three full days, all students reported that they felt comfortable controlling the Sphero balls. The students went from driving the balls to controlling the ball using block coding. The different functionalities were explained and the effects of each block were displayed in the Sphero app, by going through each tab.

C. Math Week

The second week was the start of math week. Beginning on Monday, students were asked to complete a pre-assessment which consisted of 15 questions. Students were given 45-60 minutes to complete the pre-assessment. The lessons consisted of assorted angle questions (e.g. finding the missing angle, labeling the angles, and finding the relationships between two angles), solving algebraic expressions, solving systems of equations (using substitution), and labeling shapes. After the lunch break, if the quizzes showed a high number of students struggling on a topic, the lesson plan for that day would be adjusted to their levels of understanding. Some of the activities for this week included creating and completing mazes, completing a list of shapes, and programming their own hot-potato game with the Sphero. At the end of the week, students completed the post-assessment and were given time to ask any additional questions or finish any activity of their choice from that week.

D. Science Week

The third week was dedicated to science concepts. The pre-assessment was administered on Monday of that week before lunch and students were again, allotted 45-60 minutes. The lessons for this week covered topics that included learning basic physics concepts (i.e. velocity, time, speed, distance, and acceleration), object motion concepts (Newton's Laws, inertia, momentum, force), and friction. The activities for this week also corresponded with the lessons. During this week, an example of a class day would include learning a few basic physics concepts and having a review activity. In relation to the lesson for the Sphero activity, students used formulas to determine the time and speed needed for the robot to go a certain distance and hit a designated spot. Most of the activities for this week incorporated the manipulation of formulas to complete the necessary activities and challenges.

E. Final Challenge

The final challenge was prepared for the last week and conducted on the last day of the program. It encompassed the math and science concepts learned from all three weeks of the course. With this challenge, students were encouraged to work within groups, without the assistance of the instructors and observers. The activity was designed for the different groups of students to compete against each other in different stages of the final challenge, which varied in difficulty. There were four stages in all, covering some of the most challenging, but relevant, concepts discussed during the program. All four stages of the final challenge included an assessment and an activity. The first task in each stage was a paper assessment that the students were allowed to complete as a team. The first group to answer the question(s) correctly was able to start the activity first. Each team had to answer the question(s) correctly to move on to the next activity. The four activities included in the final challenge are as follows:

- Create a shape - Students programmed their ball to make the shape listed on the activity card. First, they were given time to code the shape using the Sphero app. Once, they were ready, they were given paint to dip their ball in and were asked to run the code and place the ball on large white paper so everyone could see the shape that the painted Sphero ball created.
- Bowling - Each team took turns to see who could knock down the most pins with their ball using code only.
- Maze race - A maze was created by the instructors using tape and Legos. For this activity, the students programmed the robot to see who could get the farthest in the maze.

IV. METHOD

A. Participants

All participants were students in this summer program and a part of the Atlanta Public School System (APS). Students in the program were recruited based on the Georgia Board Rule 160-4-2-.38 [18] which says, "a gifted and talented student is defined as one who "demonstrates a high degree of intellectual and/or creative ability(ies), exhibits an exceptionally high degree of motivation, and/or excels in specific academic fields, and who needs special instruction and/or special ancillary services to achieve at levels commensurate with his or her ability". In addition, some students were financially able to attend, while others were given a scholarship.

The targeted participants were rising 6th through 8th graders. Their ages ranged from 10 to 13 years old with the average age of 11 years old. There were a total of 23 students and all of the students identified as African American except for one, who identified as White. Out of the 23 students, 14 were rising 6th graders, one student was a rising 7th grader, 4 students were rising 8th graders, and 4 did not specify. 17 of the students identified as a male and 6 as female.

B. Participants Previous Knowledge of Curriculum

For the scope of this study, we will only focus on the mathematics portion of the Atlanta Public Schools curriculum. Since the majority of the students had not yet completed 6th grade, the standards for 5th grade were added.Those who completed the 5th grade the following school year were registered as 6th graders. As mentioned earlier, most participants were on their way to 6th grade so they had yet to be exposed to any of the 6th-grade curricula. The standards for 8th grade were added to show that the Sphero program also covered some concepts that are not taught until the 8th grade. According to the Atlanta Public Schools mathematics curriculum map [18], [19]:

Upon completion of the 5th-grade, students should be able to...

- Convert like measurement units within a given measurement system.
- Geometric Measurement: understand concepts of volume and relate volume to multiplication and division.

Upon completion of the 6th grade, students should be able to...

- Apply and extend previous understandings of arithmetic to algebraic expressions.
- Reason about and solve one-variable equations and inequalities.
- Solve real-world and mathematical problems involving area, surface area, and volume.

Upon completion of the 7th-grade, students should be able to...

- Solve real-life and mathematical problems using numerical and algebraic expressions and equations.
- Analyze proportional relationships and use them to solve real-world and mathematical problems.

- Use properties of operations to generate equivalent expressions.
- Solve real-life and mathematical problems using numerical and algebraic expressions and equations.
- Draw, construct, and describe geometrical figures and describe the relationships between them.
- Solve real-life and mathematical problems involving angle measure, area, surface area, and volume.

Upon completion of the 8th-grade, students should be able to…

- Understand and apply the Pythagorean Theorem.
- Use functions to model relationships between quantities.
- Understand the connections between proportional relationships, lines, and linear equations.
- Analyze and solve linear equations and pairs of simultaneous linear equations.
- Understand the connections between proportional relationships, lines, and linear equations.
- Understand congruence and similarity using physical models, transparencies, or geometry software.

C. Research Design

The research component intended to observe middle school students' attitudes towards computing. There were two research methods used: naturalistic observation and an online survey. Before participating, parental/guardian consent forms were distributed and returned to each student. Thirteen student participants completed both the pre-survey and the post-survey in its entirety. Participants completed a pre-survey at the start of the program and a post-survey at the end. The survey consisted of four parts: (1) demographics; (2) computing attitudes; (3) career decision making; (4) and academic resilience. The demographics section collected participants' age, sex, last completed grade level, STEM grades, and future career goals. Participants were asked what computer science meant to them and to describe their previous involvement with coding and/or Sphero (if any).

Computing attitudes were investigated using an adapted version of the Subjective Science Attitude Change Measures–Student Version [16], [17], a 23-item 7-point Likert measurement scale designed to predict motivation and confidence in science. The adoption of the scale allowed science questions to remain STEM-focused but ensured students understood the term computing (computer science and computational thinking) and to incorporate the subjects while answering the questions. Career decision making confidence was examined using the 25-item, 5-point Likert scale Career Decision Making Self-Efficacy Scale-Short Form developed by Betz, Klein, and Taylor [20]. The Academic Resilience Scale developed by Cassidy [21] was used to measure students' grit during academic hardships. The Academic Resilience Scale has 30 items and uses a 5-point Likert scale to answer likelihood questions about how one would react to a scenario when one is failing a course. Naturalistic observations were coded using an inductive thematic analysis [22]. Open-ended questions in the demographic section of the pre-survey and post-survey were analyzed using an inductive-deductive thematic analysis [22]. All scales were analyzed using the recommended descriptive statistics. Three paired-sample t-tests were performed to compare the results of each scale during the pre-survey to post-survey. Individually paired sample t-tests were conducted to compare the pre-post math and science assessments.

V. RESULTS

Before each week started, students were asked if they had ever heard of the concepts or topics that would be discussed. The majority of the class claimed to not know most of the concepts mentioned. After taking both the pre-math and science assessments, the students confirmed that they did not know most, if not all, of the information presented on the assessment.

A. Pre vs Post Math Test

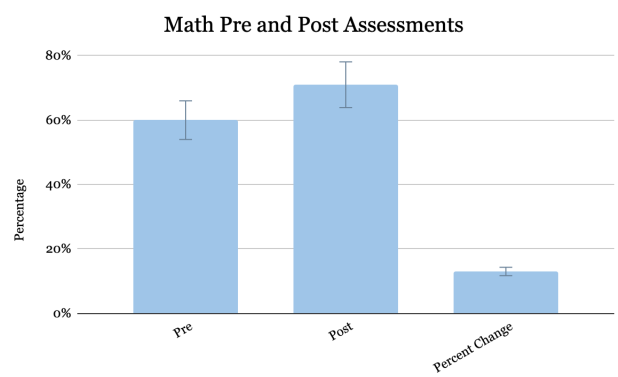

Fig. 1. Math Pre and Post-Assessments

The math assessment was first in the sequence of assessments given during the summer program. As mentioned earlier, both the pre and post-assessments consisted of 15 questions that ranged in difficulty and topic, related to mathematics. The post-assessment consisted of 15 of the same or similarly styled questions. In Figure 1, there was an improvement from the pre- to post-assessment.

The math assessment scores also showed significant growth from an initial average of about 60% to an average of about 72% for the final assessment (see Figure 1). The p-value equals 0.0146 ($t = 2.79$), which is considered statistically significant. The average percent change was about 13%. All lessons showed significant score improvements, respectively (see Figure 2).

In Figure 2, the graph displays the math pre and post-assessments based on the lesson. The assessment was split up by angles, equations, and shapes. The Angles lesson had significant score improvements, with a pre-assessment average of 68% and a post-assessment average of 88%. The Shapes lesson also had significant score improvements, with a pre-assessment average of 63% and a post average of 78%. Both p-values (0.007, $t=3.01$ and 0.048, $t=1.83$, respectively) proved

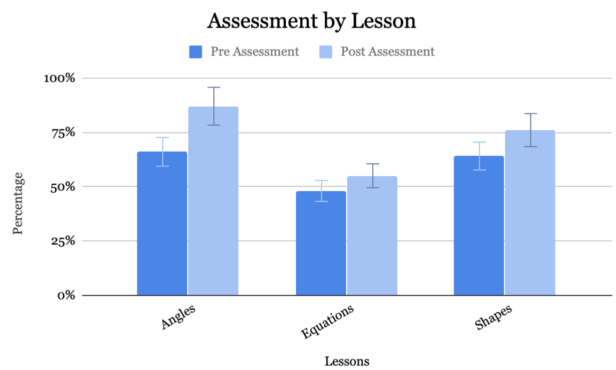

Fig. 2. Math Assessments by Lesson

to be statistically significant. For the Systems of Equations lesson, participants did not perform significantly better from the pre-assessment(45% average), to the post-assessment(58% average). The p-value equals 0.083 (t=1.49), which is also statistically significant.

1) Pre vs Post Science: The science assessment was the second assessment given. Similar to the math assessment, both the pre- and post-assessments for science ranged in difficulty and topic. As shown in Figure 3, there was considerable improvement from the pre to post-assessment.

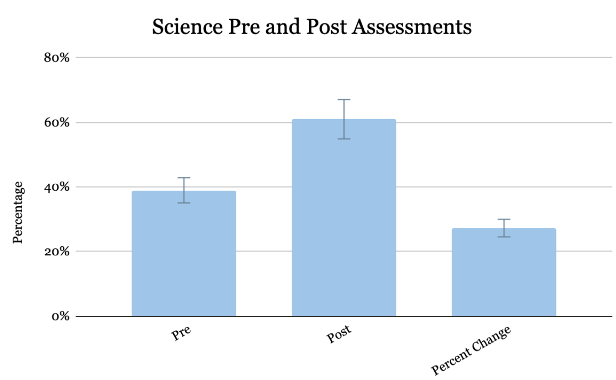

Fig. 3. Science Pre and Post Assessments

Based on this data and the students' observed reaction, they were engaged and retained quite a bit of information. The table below shows the results from the pre- and post-science assessments, in addition to the average percent change from the pre to the post. T-tests were run on all presented data and the results for both the pre and post-assessments were statistically significant, yielding a p-value of .0017. The pre-assessment had an average of about 39% with the post-assessment having an average of about 63%. On average, students did 28% better on the post-assessment and 100% of the students scored better.

Figure 4, examines the science pre and post-assessments based on the lesson. The "S, T, D, V, A" section refers to the basic physics concepts covered: speed, time, distance,

Fig. 4. Science Assessments by Lesson

velocity, and acceleration. There was a separate section on friction. The p-values for both of these sections showed to be statistically significant with the physics concepts having a p-value of 0.0172 (t = 2.73) and the friction section had a p-value of 0.0002 (t = 5.01). The "Newton, I, M, F" section refers to object motion concepts such as Newton's Laws, inertia, momentum, and force, which held a p-value of 0.0635 (t = 2.03).

Similar to the overall analysis of the pre and post-assessments, there was a 100% increase from pre to post for all sections. Based on the results, students retained the most information from the friction lesson, with an average increase of 48% from pre to post.

Observations were conducted during the assessment process. The assessment format remained constant between the math and science assessments, however, the science assessment had a few more questions. Generally, students were extremely distracted, discouraged, and/or disengaged during all assessments. As a whole, students asked more questions during both math assessments compared to both science assessments.

B. Pre/Post Survey

Thirteen of the twenty-one students completed both the pre-survey and the post-survey. A paired-samples t-test was conducted to determine the program's significance in increasing students' career decision making self-efficacy, computing attitudes, and academic resilience. A noticeable increase in career decision making self-efficacy was prevalent from the pre-survey ($M = 3.75$, $SD = 0.88$) to the post-survey ($M = 4.15$, $SD = 0.72$), $t(11) = 1.976$, $p = .037$ (upper tailed). There were no statistical differences in computing attitudes or academic resilience.

C. Observation/Reflection

There was a substantial knowledge gap with math concepts when compared to science concepts. Because of this, the instructors had to continuously stop and make sure everyone was caught up or not getting too far behind. To combat this, the instructors took a different approach. For example, when systems of equations were being taught, instructors stopped to break the students into smaller groups to teach the concepts on a more intimate level. During the reflection period, 100% of the group agreed that systems of equations were the most challenging lesson up to that point and that the students, overall did not enjoy it. The importance and the purpose of math week altogether came up during reflection often.

During science week, students were noticeably more engaged, most likely because the concepts were more applicable to their experiences. The students were more interested and enthusiastic during this week. This could be due to a number of factors, such as increased comfort levels with the instructors as well as the other students, becoming familiar with the schedule and/or incentives being introduced. Although the results of the science pre-assessments displayed students' lack of knowledge, their post-test and behavior in class showed their alertness and ability to retain the information from the week.

Throughout the weeks, there was an increase in computing jargon being used. Students began using the correct terminology when trying to explain their code. There were also connections being made when they were shown code in Python and compared it to their block coding. The instructors also used the class period to review simple Python code with the students. This code resembled the students' code and helped them to be able to identify each part of the code themselves.

Overall, the students requested more activity time and less lesson time. Towards the end of the program, they realized the role each lesson played in the activity and why it was necessary. During reflection, most stated they were glad they were taught the lessons that correspond to the activities.

VI. DISCUSSION

The summer program was designed to increase students' computing interests, attitudes, and their performance in STEM through interactive applications. Increasing students' computing attitudes is suggested to increase the likelihood that students would pursue careers in computing [12], thereby, supporting the demand for computing careers [3]. This program was successful at strengthening students' confidence in identifying and choosing careers, but it did not necessarily steer students towards computing. There was no significant change in students' attitudes and interests in computing.

The program was highly effective at increasing students' math and science comprehension and performance. To explore these findings, it is necessary to identify how the program directly addressed working with students at the middle school level. Adolescent students respond best to active learning, as active learning often incorporates peer interaction and short lesson plans to hold short-lived interests [7], [8]. Students were notably more engaged in science lessons than math lessons. Students showed more engagement for the physics science lesson than math courses due to the clear, correlating nature of the lesson and activity. This also supports literature suggestions that adolescent students respond strongly to applicable learning [7], [8]. Students were easily able to relate the physics concepts to reality during the lesson whereas math lessons tended to be a bit more abstract and related less to the activity. Consequently, a few students questioned the purpose of the math lessons. The program's use of Sphero robotics facilitated the recommended hands-on learning experience within all activities as students had to program and control their own balls as well as work in teams [9].

Literature suggests that instructing computing at an early age has positive effects on math and applied science performance and retention [23]. The findings directly support the literature, all but one lesson (Systems of Equations) significantly increased students' performances in math and science. As computer programming skills were not directly measured, it is not clear of the impact the program had on simple fundamentals of computing; however, there were findings observed regarding the students' behavior including: computing belongingness through accurate use of field-specific terminology and the ability to identify and locate important content within a few simple lines of code written in Python.

A. Limitations

There were a few limitations in the data collection of the study. Naturalistic observations were recorded, however, there was no validated framework used to deductively observe behaviors. Only thirteen of the twenty-three students participated in the summer program completed the surveys, which causes the conclusion to be less persuasive although the results were based on the calculation of t-statistic and p-value. Additionally, multiple validated scales were used to measure different academic and computing metrics; however, there was no factor analysis performed to eliminate any correlating or overlapping items. Finally, there were no test groups to independently determine the cause of the math and science performance increase. Thus, it is unclear if the increase in math and science scores was directly from the lecture, the activity, or a combination of both. It is also important to note that most students had not been introduced to the system of equations as a math subject prior to this program.

B. Significance of Small Samples and Short Programs

One of the main concerns that stems from utilizing small samples is that it would not be able to predict for the whole population, thus causing higher variability leading to results that are less reliable. However, several studies have been run that show that minorities learn and retain more information in smaller classroom settings, at all ages [24]. The overall goal of the program is to increase achievement amongst underrepresented students and to close achievement gaps between advantaged and disadvantaged students. Therefore, studies and research are in an environment similar to the overall goal. The results in the present study are significant; however, we do acknowledge that larger numbers are needed to generalize.

VII. CONCLUSION

The work performed is important,because it exposes middle school underrepresented students to computing through active learning grounded in existing STEM curricula. Though this program was not effective at increasing computing attitudes at large, it showed promising results in terms of developing students' sense of belonging in the computing field with programming and robotics comfort. The program also served as an effective STEM performance initiative that integrates much-needed computing principles to middle school students. In addition, students that participated in the program will start the year off ahead of their peers. Other research was conducted with this study focusing on developing undergraduate students into STEM near-peer instructors as well as more in-depth observation of how middle school students interact with robotics and learn primary.

A. Strengths

Overall, the Sphero summer program was successful, for both the instructors and students. The most positive result was that the students received the information quickly and retained it. As time progressed, all of the students were more comfortable talking with and asking the instructors for help if they needed further instruction or did not understand. The instructors all had different teaching methods, which gave the kids a tailored learning experience. For the students who were challenged the most, an instructor would break down the lesson to the simplest concept and build on the experiences of that student. For the students that picked up things fairly quickly, an instructor would extend the lesson to keep the fast learners engaged and connected with the group.

The instructors also did a thorough job of planning the lessons and keeping a consistent schedule. Students were able to get into a rhythm, making it easier for the instructors to move through the lesson without complications. This allowed instructors to pay closer attention to what helped the students learn. For example, the instructors realized that when the students felt more comfortable with the lesson and activities they became more creative. The instructors took the initiative to change the activities and lessons to give students more opportunities to use that creativity.

B. Implications

Even though the sample size was small, this work is important. This program, along with past similar programs, allow others to learn and build their own programs or workshops. This program was just one piece of a larger puzzle the lab is working to complete. The lab has been invited back to the same summer program to run another Sphero course with more students. In addition, many other schools, summer programs, and development centers have inquired about bringing one of the Sphero workshops to their students.

References

[1] T. Shamma, "Coding while black: Hacking the future of the tech industry," *NPR*, Jun 2016. [Online]. Available: https://www.npr.org/sections/codeswitch/2016/06/04/480630726/coding-while-black-hacking-the-future-of-the-tech-industry

[2] "Bachelor's degrees earned by african americans, by major." [Online]. Available: https://www.aps.org/programs/education/statistics/aamajors.cfm

[3] "Code.org," http://www.code.org, 2019.

[4] E. Kazakoff and M. Bers, "Programming in a robotics context in the kindergarten classroom: The impact on sequencing skills," *Journal of Educational Multimedia and Hypermedia*, vol. 21, no. 4, pp. 371–391, 2012.

[5] E. Schanzer, K. Fisler, S. Krishnamurthi, and M. Felleisen, "Transferring skills at solving word problems from computing to algebra through bootstrap," in *Proceedings of the 46th ACM Technical symposium on computer science education*. ACM, 2015, pp. 616–621.

[6] A. Eguchi, "Educational robotics for promoting 21st century skills," *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. 8, no. 1, pp. 5–11, 2014.

[7] S. D. Branting, "Hic sunt dracones-filling in the gis map for the k-12 classroom," *Journal of the Association for History and Computing Online Journal*, vol. 11, 2008.

[8] M. Ioannou and T. Bratitsis, "Utilizing sphero for a speed related stem activity in kindergarten," in *Hellenic Conference on Innovating STEM Education*, 2016.

[9] B. S. Fagin, L. D. Merkle, and T. W. Eggers, "Teaching computer science with robotics using ada/mindstorms 2.0," in *ACM SIGAda Ada Letters*, vol. 21, no. 4. ACM, 2001, pp. 73–78.

[10] W. Havice, "The power and promise of a stem education: Thriving in a complex technological world," *The overlooked STEM imperatives: Technology and engineering*, pp. 10–17, 2009.

[11] M. Resnick, "Sowing the seeds for a more creative society." *Learning & Leading with Technology*, vol. 35, no. 4, pp. 18–22, 2008.

[12] K. Gosha, T. Ridley, E. Holmes, K. Womack, and J. Scott, "Introduction to computer science for urban african american students using sphero robotics workshop," in *Proceedings of the SouthEast Conference*. ACM, 2017, pp. 252–254.

[13] K. Cannon, M. A. LaPoint, N. Bird, K. Panciera, H. Veeraraghavan, N. Papanikolopoulos, and M. Gini, "No fear: University of minnesota robotics day camp introduces local youth to hands-on technologies," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 363–368.

[14] P. Lorain, "Brain development in young adolescents, good news for middle school teachers," *National Education Association*, 2002.

[15] S. M. Hadfield, J. T. Raynor, and M. D. Sievers, "Engaging secondary and post-secondary students to learn and explore programming using a theme-based curriculum and the sphero sprk+ robot," in *Proceedings of the 23rd Western Canadian Conference on Computing Education*, ser. WCCCE '18. New York, NY, USA: ACM, 2018, pp. 2:1–2:6. [Online]. Available: http://doi.acm.org/10.1145/3209635.3209643

[16] M. K. R. Stake, Jayne E., "Science enrichment programs for gifted high school girls and boys: Predictors of program impact on science confidence and motivation," *Journal of Research in Sciwence Teaching*, vol. 38, no. 10, pp. 1065–1088, 2001.

[17] E. D. Deemer, J. L. Smith, D. B. Thoman, and J. P. Chase, "Precision in career motivation assessment: Testing the subjective science attitude change measures," *Journal of Career Assessment*, vol. 22, no. 3, pp. 489–504, 2014.

[18] "Mathematics georgia standards of excellence (gse) k-5." [Online]. Available: https://www.georgiastandards.org/Georgia-Standards/Pages/Math-K-5.aspx

[19] "Mathematics / curriculum maps." [Online]. Available: https://www.atlantapublicschools.us/Page/50330

[20] N. E. Betz, K. L. Klein, and K. M. Taylor, "Evaluation of a short form of the career decision-making self-efficacy scale," *Journal of Career Assessment*, vol. 4, no. 1, pp. 47–57, 1996.

[21] S. Cassidy, "The academic resilience scale (ars-30): A new multidimensional construct measure," *Frontiers in psychology*, vol. 7, p. 1787, 2016.

[22] J. Fereday and E. Muir-Chochrane, "Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development," *International Journal of Qualitative Methods*, vol. 5, no. 1, pp. 80–92, 2006.

[23] A. Sullivan and M. U. Bers, "Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade," *International Journal of Technology and Design Education*, vol. 26, no. 1, pp. 3–20, 2016.

[24] B. Nye, L. V. Hedges, and S. Konstantopoulos, "Do minorities experience larger lasting benefits from small classes?" *The Journal of Educational Research*, vol. 98, no. 2, pp. 94–100, 2004.